

Mise en place de GLPI sous Debian

- [Installation de Debian](#)
- [Installation des prérequis pour GLPI](#)
- [Installation de GLPI](#)
- [Installation du plugin GLPI Inventory](#)

Installation de Debian

Pour ce projet, j'ai choisi d'utiliser **Debian Bookworm (12.5.0)** sur une machine virtuelle tournant sous Linux KVM avec Qemu-KVM.

J'ai commencé par télécharger l'image ISO de Debian Bookworm depuis le site officiel. Ensuite, j'ai créé une nouvelle machine virtuelle en utilisant virt-manager. Pour cette VM, je l'ai nommée **GLPI** et j'ai configuré les paramètres suivants :

- **Type de système d'exploitation** : Linux
- **Version** : Debian 12
- **Disque dur** : 45 GB
- **Mémoire** : 4 GB
- **Processeurs** : 2 (virtuels)

J'ai ajouté l'image ISO téléchargée dans le lecteur virtuel de CD/DVD pour démarrer l'installation et configuré l'interface réseau en mode NAT. J'ai ensuite lancé la machine virtuelle et suivi les étapes d'installation de Debian, notamment la sélection de la langue, de la région et la configuration du clavier.

J'ai choisi de ne pas installer d'interface graphique et d'ajouter le serveur web et le serveur SSH lors de l'installation

Après le redémarrage post installation, je me suis connecté sur le tty avec les identifiants que j'ai fourni durant la phase d'installation de Debian

J'ai ensuite tapé la commande :

```
hostname -I
```


```
root@glpi:~# hostname -I  
192.168.122.66
```

Afin d'obtenir l'adresse IP qui a été fournie à ma machine virtuelle dans le but de m'y connecter en SSH et avoir un affichage du terminal un peu plus confortable pour la suite

Pour me connecter en SSH j'ai utilisé le client Termius, j'ai entré les informations de ma machine virtuelle (IP, Identifiants, port ssh...) dans l'application

```
root@glpi:~# neofetch
_,met$$$$$gg.
,g$$$$$$$$$$$$$P.
,g$$$P"      ""Y$$.
,$$P'        `$$$
',$$P      ,ggs.  `$$b:
`d$$'      ,P"'   $$
$$P        d$'    $$P
$$:        $$.    -   ,d$$'
$$;        Y$b._   _ ,d$P'
Y$$.      `."Y$$$$P"'
`$$b      "-._
`Y$$
`Y$$
`$$b.
`Y$b.
`"Y$b._
`""""

root@glpi
-----
OS: Debian GNU/Linux 12 (bookworm) x86_64
Host: KVM/QEMU (Standard PC (Q35 + ICH9, 200
Kernel: 6.1.0-21-amd64
Uptime: 15 hours, 51 mins
Packages: 460 (dpkg)
Shell: bash 5.2.15
Resolution: 1280x800
CPU: 11th Gen Intel i3-1115G4 (2) @ 2.995GHz
GPU: 00:01.0 Red Hat, Inc. Virtio 1.0 GPU
Memory: 135MiB / 3915MiB
```



Tout fonctionne super ! Je peux maintenant passer a l'installation et a la configuration de GLPI

Installation des prérequis pour GLPI

Une fois l'installation de Debian terminée, je me rends sur le site de GLPI afin de voir les prérequis pour qu'il fonctionne

La documentation officielle de GLPI m'informe qu'il me faut :

- **Un serveur Web** (Comme j'ai choisi d'installer un serveur web a l'installation de Debian, je dispose déjà de Apache 2.4.59)
- **PHP** avec comme version minimum 7.4
- **Une base de données**, la documentation de GLPI m'indique que uniquement MySQL et MariaDB sont supportées par GLPI, j'ai choisi d'utiliser MariaDB (pour changer de mon installation Windows ou j'avais utilisé MySQL)



Mise en place d'une IP Fixe

Avant de faire quoi que ce soit, je m'assure que ma machine dispose d'une adresse IP fixe. J'effectue la commande `ip a` dans le but d'obtenir le nom actuel de mon interface réseau.

```
root@glpi:/home# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 52:54:00:48:a4:3e brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.26/24 brd 192.168.122.255 scope global dynamic enp1s0
        valid_lft 3440sec preferred_lft 3440sec
    inet6 fe80::5054:ff:fe48:a43e/64 scope link
```

```
valid_lft forever preferred_lft forever
```

Ici le nom de mon interface est **enp1s0**.

Je consulte ensuite le fichier `/etc/network/interfaces`.

```
This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

```
source /etc/network/interfaces.d/*
```

```
# The loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
# The primary network interface
```

```
allow-hotplug enp1s0
```

```
iface enp1s0 inet dhcp
```

Je vois que mon interface est actuellement en **DHCP**, je passe donc l'adressage en **fixe** en modifiant ce fichier de cette manière :

```
iface enp1s0 inet static
    address 192.168.122.26
    netmask 255.255.255.0
    gateway 192.168.122.1
    dns-nameservers 8.8.8.8 8.8.4.4
```

Je lui ai tout simplement "Fixé" l'adresse qu'il avait déjà, je n'ai plus qu'à redémarrer le service réseau afin d'appliquer mes modifications avec la commande `systemctl restart networking`

Il faut maintenant que je fasse une exclusion ou une réservation sur mon serveur DHCP. Dans mon étendue DHCP, j'indique qu'il faut exclure l'adresse **192.168.122.26** de la plage.

Installation des paquets manquants

Pour installer les paquets manquants, je m'assure d'abord que tout est à jour en faisant la commande

```
apt update && apt upgrade -y
```

J'installe ensuite les deux paquets qui me manquent c'est-à-dire **PHP** et la base de données **MariaDB**.

```
apt install php mariadb-server
```

Les dépôts de Debian m'installent PHP 8.2 ainsi que MariaDB 10.11.6.

Configuration initiale de PHP

Dans la documentation de GLPI, on nous indique qu'il faut activer des extensions obligatoires dans le bon fonctionnement du service.

“ Ces extensions PHP suivantes sont requises pour que l'application fonctionne correctement :

- `dom`, `fileinfo`, `filter`, `libxml`, `json`, `simplexml`, `xmlreader`, `xmlwriter`: these PHP extensions are enable by default and are used for various operations;
- `curl`: used for remote access to resources (inventory agent requests, marketplace, RSS feeds, ...);
- `gd`: used for images handling;
- `intl`: used for internationalization;
- `mysqli`: used for database connection;
- `session`: used for sessions support;
- `zlib`: used for handling of compressed communication with inventory agents, installation of gzip packages from marketplace and PDF generation.

Je vais donc éditer le fichier `php.ini` présent dans le répertoire `/etc/php/8.2/apache2/`

Je me rends dans la catégorie "Dynamic Extensions" et enlève-le `;` devant chaque extension que je souhaite activer.

Configuration de la base de données MariaDB



Avant de faire quoi que ce soit, je lance le script :

```
mysql_secure_installation
```

Il me permet de définir plusieurs options de sécurité comme supprimer le mot de passe root et définir les options de connexion à distance, etc.

```
root@glpi:~# mysql_secure_installation
```

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):

OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n

... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n

... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y

... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n

... skipping.

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n] y

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n] y

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

Je vais ensuite créer une base de données dédiée à GLPI, je me connecte donc en tant que superutilisateur à MariaDB.

```
mysql -u root -p
```

Je crée ma base de données avec comme nom "glpi".

```
CREATE DATABASE glpi;
```

Je crée un utilisateur "**administrateurglpi**" avec comme mot de passe "**motdepasse**"

```
CREATE USER 'administrateurglpi'@'localhost' IDENTIFIED BY 'motdepasse';
```

Je donne tous les droits à mon utilisateur sur la base de données **glpi**


```
GRANT ALL PRIVILEGES ON glpi.* TO 'administrateurglpi'@'localhost';
```

Et je recharge la table des privilèges pour appliquer mes modifications et je quitte

```
FLUSH PRIVILEGES;  
EXIT;
```

Je dispose désormais de la configuration suivante :

Base de données : glpi

Utilisateur ayant les droits sur la base de données : administrateurglpi

Mot de passe de l'utilisateur : motdepasse

Adresse de la base de données : localhost/192.168.122.26

Configuration d'Apache



Il faut que je configure un hôte virtuel dans Apache pour mon GLPI. Un hôte virtuel me permettra d'ajouter plusieurs services plus tard sur le même serveur Apache si je le souhaite.

Je crée le fichier de configuration pour GLPI

```
nano /etc/apache2/sites-available/glpi.conf
```

Et j'ajoute la configuration recommandée par la documentation GLPI dans ce fichier.

```
<VirtualHost *:80>  
    ServerName glpi.localhost  
  
    DocumentRoot /var/www/glpi/public  
  
    # If you want to place GLPI in a subfolder of your site (e.g. your virtual host is serving multiple applications),  
    # you can use an Alias directive. If you do this, the DocumentRoot directive MUST NOT target the GLPI  
    directory itself.  
    # Alias "/glpi" "/var/www/glpi/public"
```

```
<Directory /var/www/glpi/public>
    Require all granted

    RewriteEngine On

    # Ensure authorization headers are passed to PHP.
    # Some Apache configurations may filter them and break usage of API, CalDAV, ...
    RewriteCond %{HTTP:Authorization} ^(.+)$
    RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

    # Redirect all requests to GLPI router, unless file exists.
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^(.*)$ index.php [QSA,L]
</Directory>
</VirtualHost>
```

J'active le module rewrite, car il est demandé dans le fichier de configuration (Ligne 13).

```
a2enmod rewrite
```

J'active le site que je viens de créer.

```
a2ensite glpi.conf
```

J'en profite pour désactiver le site par défaut.

```
a2dissite 000-default.conf
```

Je redémarre ensuite le service apache2

```
systemctl restart apache2
```

Téléchargement de GLPI

Il ne me reste plus qu'à télécharger GLPI et à le placer à la racine de mon serveur web, pour faire cela, je me rends sur le site officiel de GLPI afin de récupérer l'URL de téléchargement et de télécharger l'archive avec wget

```
wget https://github.com/glpi-project/glpi/releases/download/10.0.15/glpi-10.0.15.tgz
```

J'extrait ensuite l'archive dans le répertoire **/var/www**

```
tar -xvf glpi-10.0.15.tgz -C /var/www/
```

Et donne les permissions a Apache d'accéder au répertoire /var/www/glpi

```
chown -R www-data:www-data /var/www/glpi/  
chmod -R 755 /var/www/glpi/
```

Création des dossiers de configuration

La documentation de GLPI nous indique qu'il faut que la configuration de GLPI soit stockée dans **/etc/glpi** et déplacer la configuration actuelle dans celui-ci. Je donne également les permissions d'accès au répertoire

```
mkdir /etc/glpi  
mv /var/www/glpi/config/ /etc/glpi/  
chown -R www-data /etc/glpi/
```

Il nous indique qu'il faut faire pareil avec le contenu du dossier files dans **/var/lib/glpi/**

```
mkdir /var/lib/glpi  
mv /var/www/glpi/files/ /var/lib/glpi/  
chown -R www-data /var/lib/glpi/
```

Il faudra aussi créer un dossier **/var/log/glpi** afin d'y accueillir les logs, rien à copier ici

```
mkdir /var/log/glpi  
chown -R www-data /var/log/glpi/
```

Il faudra également créer un fichier **inc/downstream.php** dans le dossier de GLPI avec le contenu indiqué sur la documentation de GLPI

```
nano /var/www/glpi/inc/downstream.php
```

Et colle le contenu suivant

```
<?php  
define('GLPI_CONFIG_DIR', '/etc/glpi/');  
  
if (file_exists(GLPI_CONFIG_DIR . '/local_define.php')) {  
    require_once GLPI_CONFIG_DIR . '/local_define.php';  
}
```

Je fais la même chose pour le fichier `/etc/glpi/local_define.php`

```
nano /etc/glpi/local_define.php
```

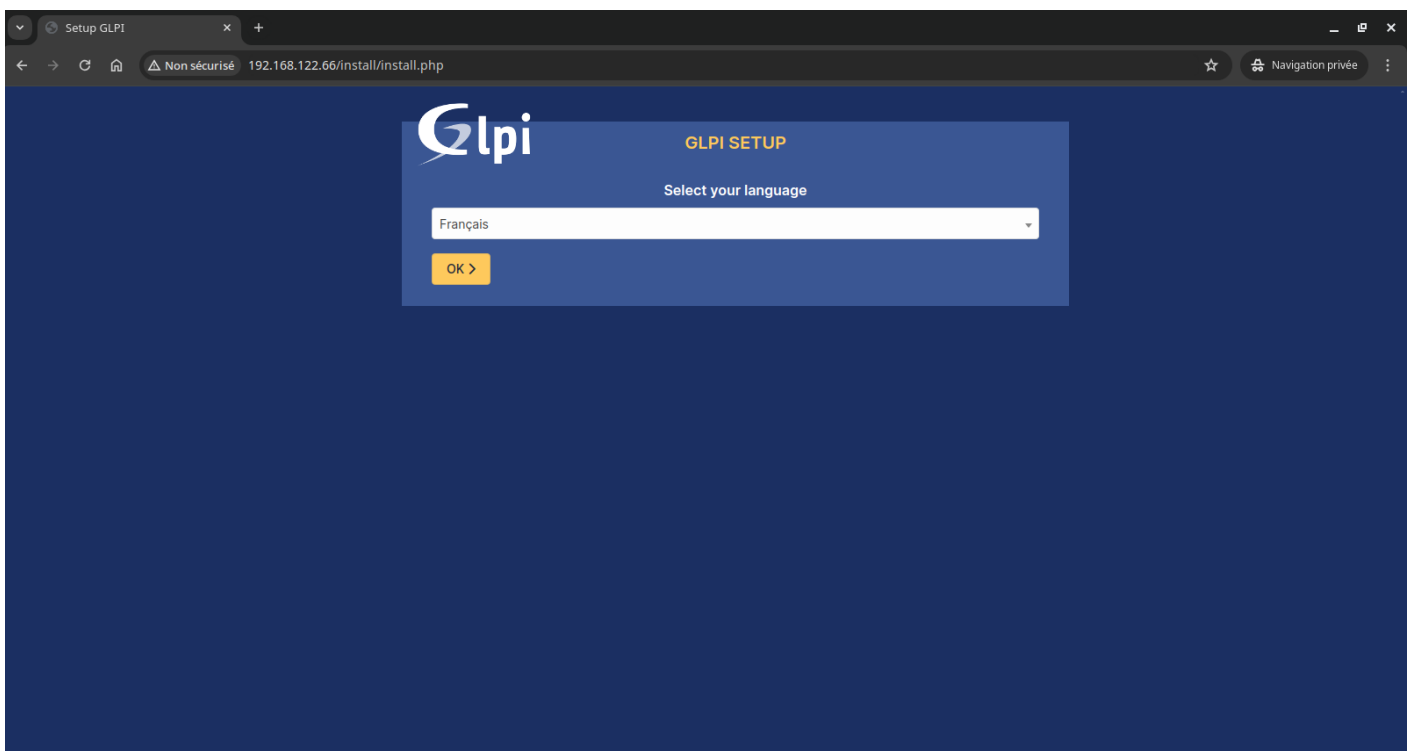
Et je colle le contenu suivant

```
<?php  
define('GLPI_VAR_DIR', '/var/lib/glpi');  
define('GLPI_LOG_DIR', '/var/log/glpi');
```

Vérification de la configuration



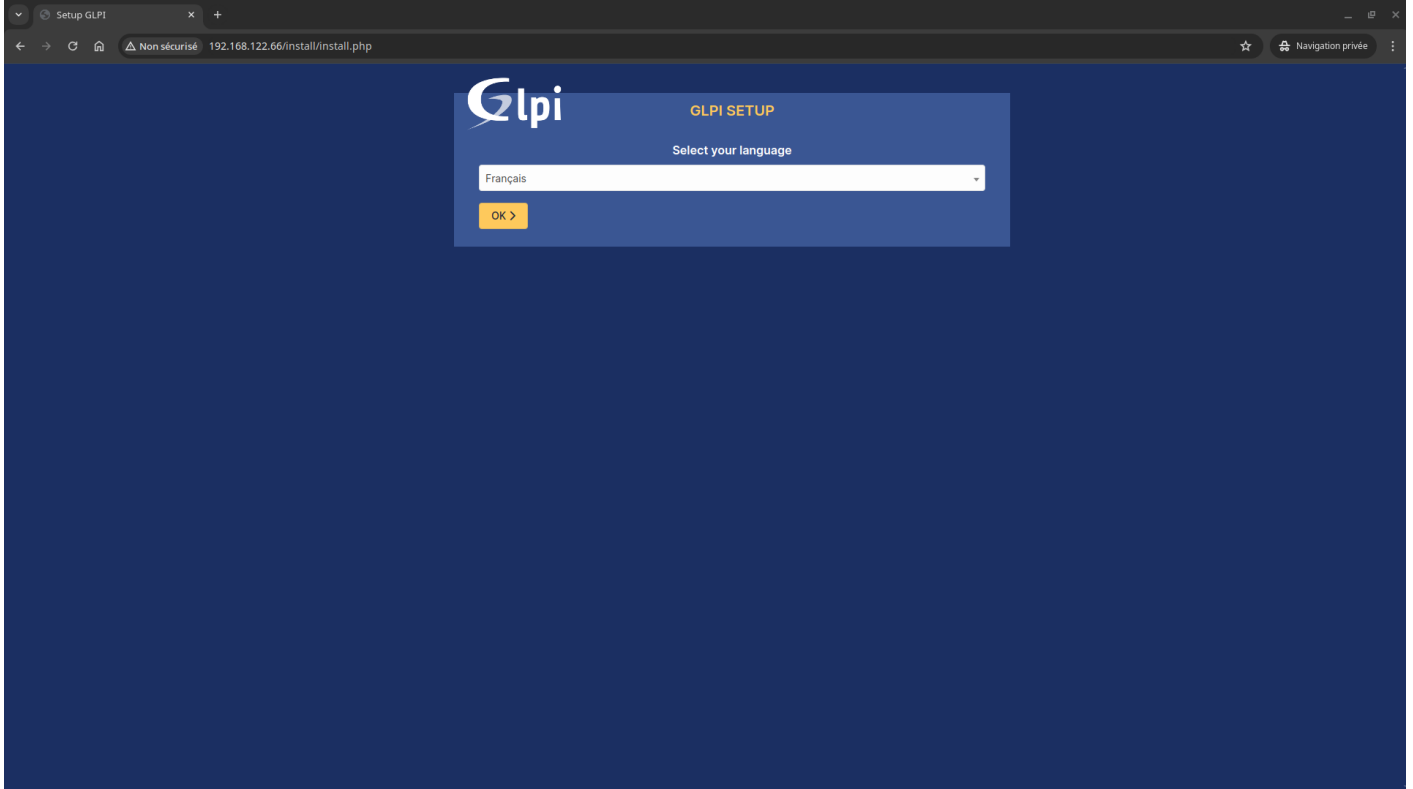
Pour vérifier que j'ai bien tout configuré, je tape l'adresse IP de ma machine virtuelle dans mon navigateur internet.



J'arrive bien sur la page d'installation de GLPI je peux maintenant passer à l'étape suivante !

Installation de GLPI

Super ! Je suis maintenant dans l'assistant d'installation de GLPI



Je choisis la langue de l'interface et je continue en cliquant sur "**Ok**"

J'arrive sur la page de licence, je la lis et je clique sur "**Continuer**"



GLPI SETUP

Licence

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

[Des traductions non officielles sont également disponibles](#)

Continuer >

L'utilitaire me demande si je souhaite effectuer une installation ou bien une mise à jour. Etant donné qu'il s'agit d'une Installation, je sélectionne **"Installer"**



GLPI SETUP

Début de l'installation



Installation ou mise à jour de GLPI

Choisissez 'Installation' pour une nouvelle installation de GLPI.

Choisissez 'Mise à jour' pour lancer la mise à jour de votre version de GLPI à partir d'une version antérieure.

Installer

Mettre à jour

Oh ! GLPI m'informe qu'il a besoin des droits d'accès sur les fichiers `/etc/glpi/config_db.php`,
`/etc/glpi/glpicrypt.key`



GLPI SETUP

Étape 0

Vérification de l'accès en écriture sur les fichiers de configuration

❗ **Accès en écriture refusé sur les fichiers de configuration**

Un accès temporaire en écriture est nécessaire pour les fichiers suivants : `/etc/glpi//config_db.php`, `/etc/glpi//glpicrypt.key`.

L'accès en écriture sur ces fichiers pourra être supprimé une fois l'opération terminée.

Voulez-vous continuer ?

Réessayer ↺

Je retourne donc dans ma console SSH afin de donner le droit d'accès temporaire à ces fichiers

```
chmod o+w /etc/glpi/
```

Dans la page suivante, GLPI m'indique que tout est bon, bien qu'une alerte apparaisse au niveau de l'extension ldap qui n'est pas activée, je n'ai pas besoin de cette extension donc je peux ignorer et cliquer sur "**Continuer**"

Requis	Intl extension <i>Requis pour l'internationalisation.</i>	✓
Requis	zlib extension <i>Requis pour la gestion de la communication compressée avec les agents d'inventaire, l'installation de paquets gzip à partir du Marketplace et la génération de PDF.</i>	✓
Requis	Libsodium ChaCha20-Poly1305 constante de taille <i>Activer l'utilisation du cryptage ChaCha20-Poly1305 requis par GLPI. Il est fourni par libsodium à partir de la version 1.0.12.</i>	✓
Requis	Permissions pour les fichiers de log	✓
Requis	Permissions pour les dossiers de données	✓
Sécurité	Version de PHP maintenue <i>Une version de PHP maintenue par la communauté PHP devrait être utilisée pour bénéficier des correctifs de sécurité et de bogues de PHP.</i>	✓
Sécurité	Configuration sécurisée du dossier racine du serveur web <i>La configuration du dossier racine du serveur web devrait être <code>/var/www/glpi/public</code> pour s'assurer que les fichiers non publics ne peuvent être accessibles.</i>	✓
Sécurité	Configuration de sécurité pour les sessions <i>Permet de s'assurer que la sécurité relative aux cookies de session est renforcée.</i>	✓
Suggéré	Taille d'entier maximal de PHP <i>Le support des entiers 64 bits est nécessaire pour les opérations relatives aux adresses IP (inventaire réseau, filtrage des clients API, ...).</i>	✓
Suggéré	exif extension <i>Renforcer la sécurité de la validation des images.</i>	✓
Suggéré	ldap extension <i>Active l'utilisation de l'authentification à un serveur LDAP distant. L'extension ldap n'est pas présente.</i>	⚠
Suggéré	openssl extension <i>Active l'envoi de courriel en utilisant SSL/TLS.</i>	✓
Suggéré	Extensions PHP pour le marketplace <i>Permet le support des formats de paquets les plus communs dans le marketplace.</i>	✓
Suggéré	Zend OPcache extension <i>Améliorer les performances du moteur PHP.</i>	✓
Suggéré	Extensions émulées de PHP <i>Améliorer légèrement les performances.</i>	✓
Suggéré	Permissions pour le répertoire du marketplace <i>Active l'installation des plugins à partir du Marketplace.</i>	✓
<p>Voulez-vous continuer ?</p> <p>Continuer > Réessayer ↺</p>		

L'assistant d'installation de GLPI me demande ensuite les identifiants de mon service MariaDB, j'indique les informations du service que j'ai configuré plus tôt.



GLPI SETUP

Étape 1

Configuration de la connexion à la base de données

Serveur SQL (MariaDB ou MySQL)

localhost

Utilisateur SQL

administrateurglpi

Mot de passe SQL

.....

Continuer >

Sur la page d'après je choisis la base de données que j'ai crée lors de la configuration de MariaDB



GLPI SETUP

Étape 2

Test de connexion à la base de données



Connexion à la base de données réussie

Veuillez sélectionner une base de données :

Créer une nouvelle base ou utiliser une base existante :



glpi

Continuer >

Après avoir cliqué sur "**Continuer**" l'assistant initialise la base de données



GLPI SETUP

Étape 3

Initialisation de la base de données.

OK - La base a bien été initialisée

Continuer >

Sur la page d'après, GLPI me propose d'envoyer des données de télémétrie pour faire des statistiques d'usage de GLPI, je décide de laisser cette fonction activée



GLPI SETUP

Étape 4

Récolter des données

☒ Envoyer "statistiques d'usage"

Nous avons besoin de vous pour améliorer GLPI et son écosystème de plugins !

Depuis GLPI 9.2, nous avons introduit une nouvelle fonctionnalité de statistiques appelée "Télémétrie", qui envoie anonymement, avec votre permission, des données à notre site de télémétrie.

Une fois envoyées, les statistiques d'usage sont agrégées et rendues disponibles à une large audience de développeurs GLPI.

Dites-nous comment vous utilisez GLPI pour que nous améliorons GLPI et ses plugins !

[Voir ce qui serait envoyé...](#)

Référez votre GLPI

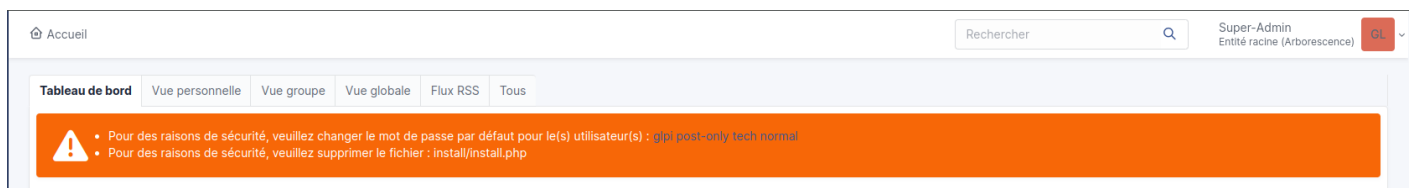
Par ailleurs, si vous appréciez GLPI et sa communauté, prenez une minute pour référencer votre organisation en remplissant le formulaire suivant [Le formulaire d'inscription](#)

Continuer >

La page suivante affiche des informations complémentaires



Sur la page d'accueil, GLPI m'indique qu'il est extrêmement conseillé de changer les mots de passe par défaut des différents comptes pour des raisons de sécurité, il est aussi conseillé de supprimer le fichier install.php



C'est tout bon ! L'installation de GLPI est terminée ! Mon GLPI est désormais opérationnel et prêt à être approvisionné

Je n'oublie pas de retirer l'autorisation temporaire mise sur le fichier **/etc/glpi** pendant l'installation

```
chmod o-w /etc/glpi/
```

Installation du plugin GLPI Inventory

Maintenant que j'ai un GLPI fonctionnel, je peux ajouter le plugin GLPI Inventory

Ce plugin permet d'automatiser et de faire de la découverte réseau de différents équipements soit via des protocoles comme SNMP ou bien via des machines ayant un "agent" (un logiciel) installé destiné à envoyer les informations relatives au matériel, logiciel, réseau de celle-ci

Téléchargement du Plugin

Pour télécharger le Plugin, je me rend sur la page Github de celui-ci : <https://github.com/glpi-project/glpi-inventory-plugin>

Je me dirige ensuite dans les releases du projet à droite afin d'avoir la dernière version du projet. Je copie l'adresse de téléchargement de la dernière version et je me rends sur mon client ssh afin de faire la commande

```
wget https://github.com/glpi-project/glpi-inventory-plugin/releases/download/1.3.5/glpi-glpinventory-1.3.5.tar.bz2
```

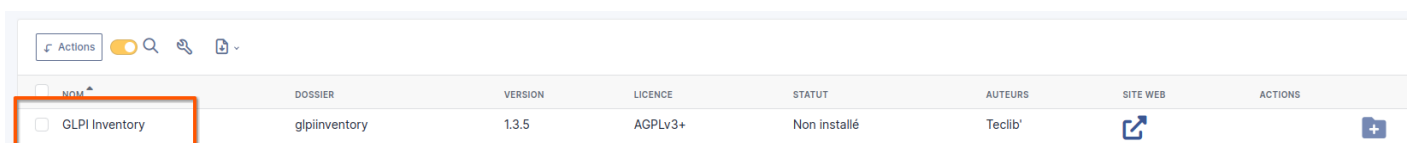
Je me retrouve avec une archive appelée **glpi-glpinventory-1.3.5.tar.bz2**

Je n'ai plus qu'à l'extraire dans le répertoire **/var/www/glpi/plugins/**

```
tar -xvf glpi-glpinventory-1.3.5.tar.bz2 -C /var/www/glpi/plugins/
```

Activation et configuration du Plugin

Dans mon navigateur, dans l'interface de GLPI je me rend dans **"Configuration"** et ensuite dans **"Plugins"**



NOM	DOSSIER	VERSION	LICENCE	STATUT	AUTEURS	SITE WEB	ACTIONS
<input type="checkbox"/> GLPI Inventory	glpiinventory	1.3.5	AGPLv3+	Non installé	Teclib'	🔗	+

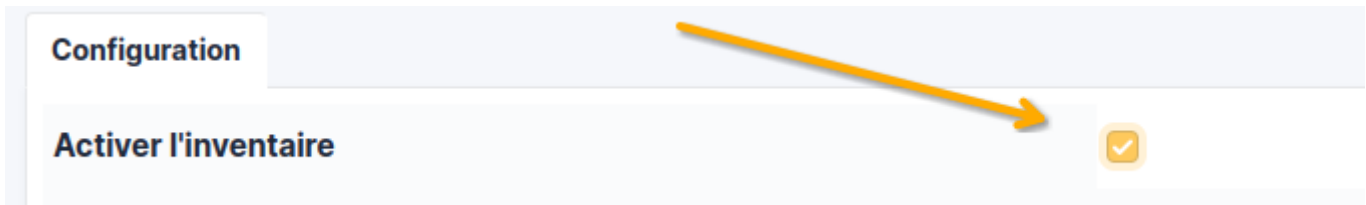
Mon plugin est bien présent ! Je peux cliquer sur l'icône en forme de dossier avec un + pour l'installer

Une fois le plugin installé, une notification me demande si je veux l'activer, comme je veux l'activer

je clique sur "Activer"



Je me rend ensuite dans l'onglet "**Administration**" et ensuite "**Inventaire**" et j'active l'inventaire



Tests de la configuration de GLPI Inventory

Il existe une application appelée GLPI Agent me permettant d'interagir avec le plugin GLPI Inventory et d'ajouter mon ordinateur dans l'inventaire : <https://github.com/glpi-project/glpi-agent/>

Je vais ajouter ma machine virtuelle ou tourne GLPI dans l'inventaire
Comme ma machine virtuelle est actuellement sous Linux je me rends dans "Releases" a droite une nouvelle fois et je prends l'installateur Linux dans la catégorie "**Linux Installer**" : **glpi-agent-1.8-linux-installer.pl**

```
wget https://github.com/glpi-project/glpi-agent/releases/download/1.8/glpi-agent-1.8-linux-installer.pl
```

Je lance ensuite le script avec perl

```
perl glpi-agent-1.8-linux-installer.pl
```

Le script me demande d'entrer l'url du serveur GLPI, j'entre donc l'url de mon serveur

```
Installing glpi-agent v1.8...
glpi-agent is about to be installed as service

Provide an url to configure GLPI server:
> http://glpi.local/

Provide a path to configure local inventory run or leave it empty:
```

>

Provide a tag to configure or leave it empty:

>

Applying configuration...

Enabling glpi-agent service...

Je lance ensuite la commande `glpi-agent` afin de ne pas attendre et de lancer l'inventaire immédiatement

```
root@glpi:~# glpi-agent
[info] target server0: server http://glpi.local/
[info] sending prolog request to server0
[info] server0 answer shows it supports GLPI Agent protocol
[info] running task Inventory
[info] New inventory from glpi-2024-05-22-15-37-50 for server0
```

Je peux voir désormais que si je me rend dans "**Parc**" et ensuite dans "**Ordinateurs**" ma machine virtuelle apparaît bien !

NOM		NUMÉRO DE		SYSTÈME D'EXPLOITATION		DERNIÈRE		COMPOSANTS	
	STATUT	FABRICANT	SÉRIE	TYPE	MODÈLE	- NOM	LIEU	MODIFICATION	PROCESSEUR
<input type="checkbox"/>	glpi	QEMU		QEMU	Standard PC (Q35 + ICH9, 2009)	Debian GNU/Linux 12 (bookworm)		2024-05-22 13:40	11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz

L'inventaire se fera maintenant automatiquement **toutes les 24h**, si l'ordinateur n'as pas fait d'inventaire durant les dernières 24h car il était éteint par exemple, il fera un nouvel inventaire **dans l'heure**